



Instituto de Ciência e Tecnologia
Universidade Federal de São Paulo

Compiladores:

Análise Léxica - De expressões regulares para AFND

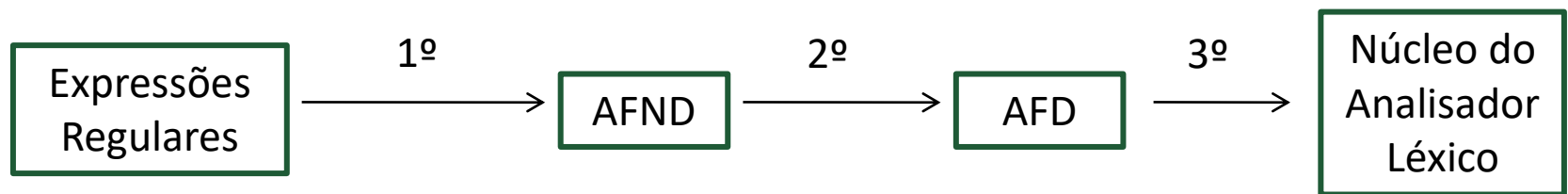
Profª Thaína A. A. Tosta

tosta.thaina@unifesp.br

São José dos Campos – 2021/2

De expressões regulares para AFND

- Expressões regulares são compactas e preferíveis como descrições de *tokens*, se comparadas com AFDs;
- O processo de construção do analisador léxico consiste de três passos:



- Geradores de analisadores léxicos seguem esse modelo.

De expressões regulares para AFND

Ao discutir **análise léxica**, usamos três termos relacionados:

- ***Token***: o nome do *token* é um rótulo que identifica um tipo de unidade léxica;
- **Lexema**: sequência de caracteres no programa fonte que casa com um *token*, sendo identificada pelo analisador léxico como uma instância de um *token*;
- **Padrão**: é uma descrição da forma que os lexemas de um *token* podem assumir, comumente definida por expressões regulares.

De expressões regulares para AFND

TOKEN	DESCRIÇÃO INFORMAL	EXEMPLOS DE LEXEMAS
if	caracteres i, f	if
else	caracteres e, l, s, e	else
comparison	< or > ou <= ou >= ou == ou !=	<=, !=
id	letra seguida por letras e dígitos	pi, score, D2
number	qualquer constante numérica	3.14159, 0, 6.02e23
literal	qualquer caractere diferente de ", cercado por "	"core dumped"

FIGURA 3.2 Exemplos de tokens.

De expressões regulares para AFND

Algoritmo de Thompson

Compõe um autômato finito não determinístico pela combinação de pequenos autômatos que reconhecem os elementos primitivos de um expressão regular:

- Um símbolo do alfabeto da linguagem;
- Concatenação de duas expressões regulares;
- Alternativa de duas expressões regulares;
- Repetição (zero ou mais vezes) de uma expressão regular.

ϵ - Expressão regular que representa a *string* vazia.

De expressões regulares para AFND

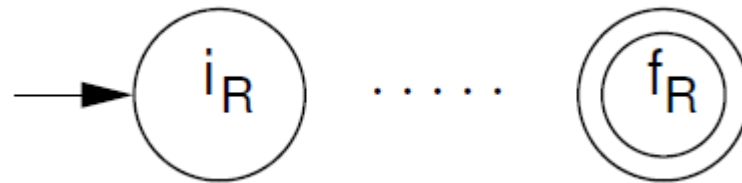
Algoritmo de Thompson

- É um método indutivo;
- O AFND é construído com base na transição pela *string* vazia que permite a junção das subexpressões;
- O algoritmo é baseado em passos muito simples, o que facilita o processo de automatização do algoritmo.

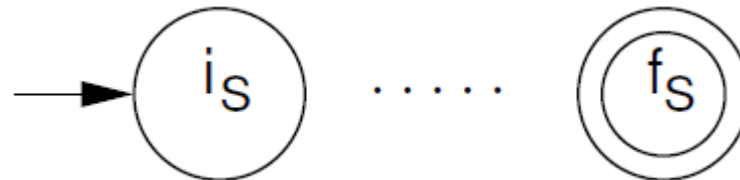
De expressões regulares para AFND

Algoritmo de Thompson

- Autômato para a expressão regular R



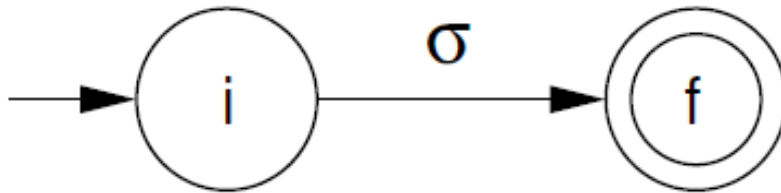
- Autômato para a expressão regular S



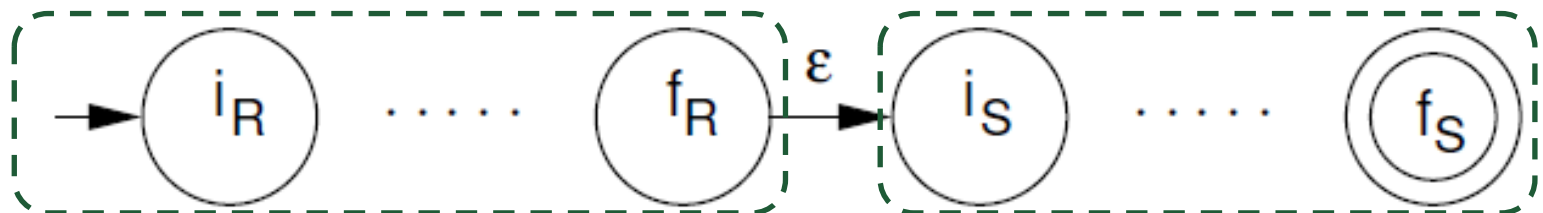
De expressões regulares para AFND

Algoritmo de Thompson

- Autômato que reconhece um símbolo do alfabeto



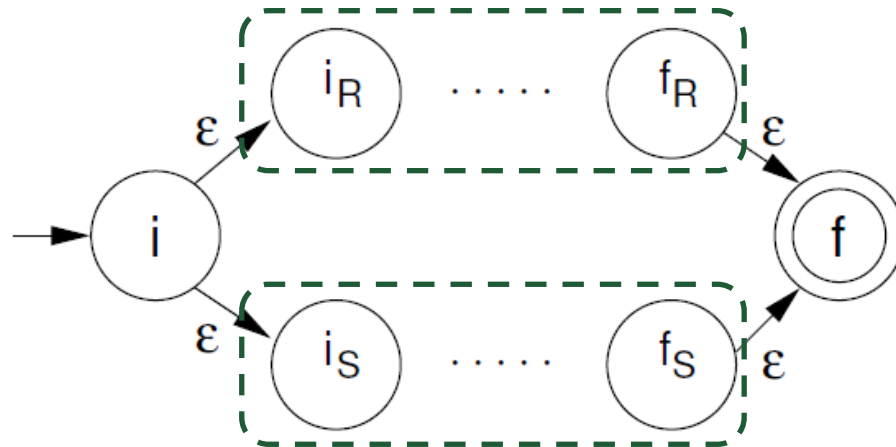
- Autômato que reconhece RS



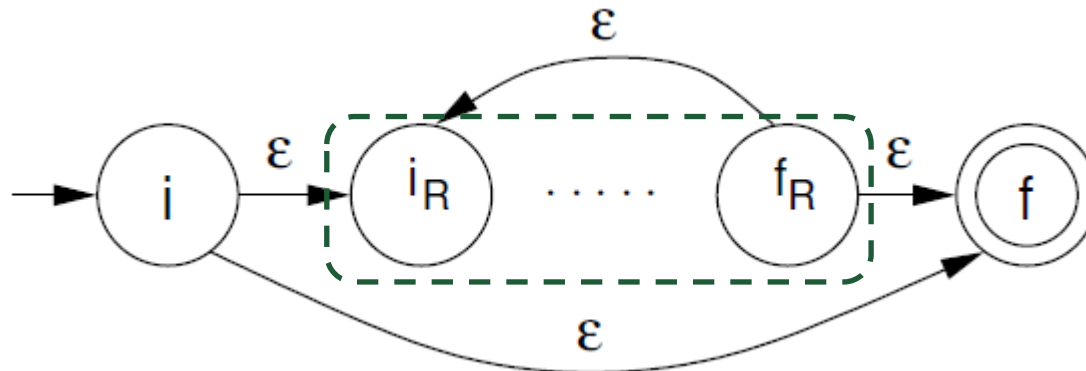
De expressões regulares para AFND

Algoritmo de Thompson

- Autômato que reconhece $R|S$



- Autômato que reconhece R^*



De expressões regulares para AFND

Algoritmo de Thompson

Exemplo: AFND para a expressão regular $(0|1)^*0$

Exemplos de *strings* válidas:

00	10
010	110
0000	1100

Exemplos de *strings* não válidas:

01	11
011	101
0001	1101

De expressões regulares para AFND

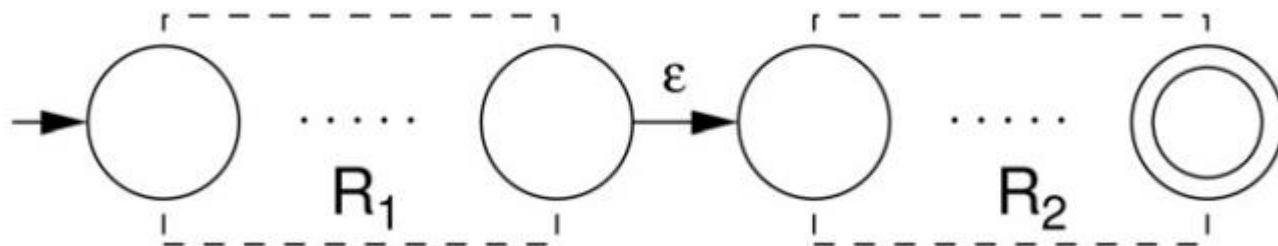
Algoritmo de Thompson

$$R = (0|1)^*0$$

$$R = R_1R_2$$

$$R_1 = (0|1)^*$$

$$R_2 = 0$$



De expressões regulares para AFND

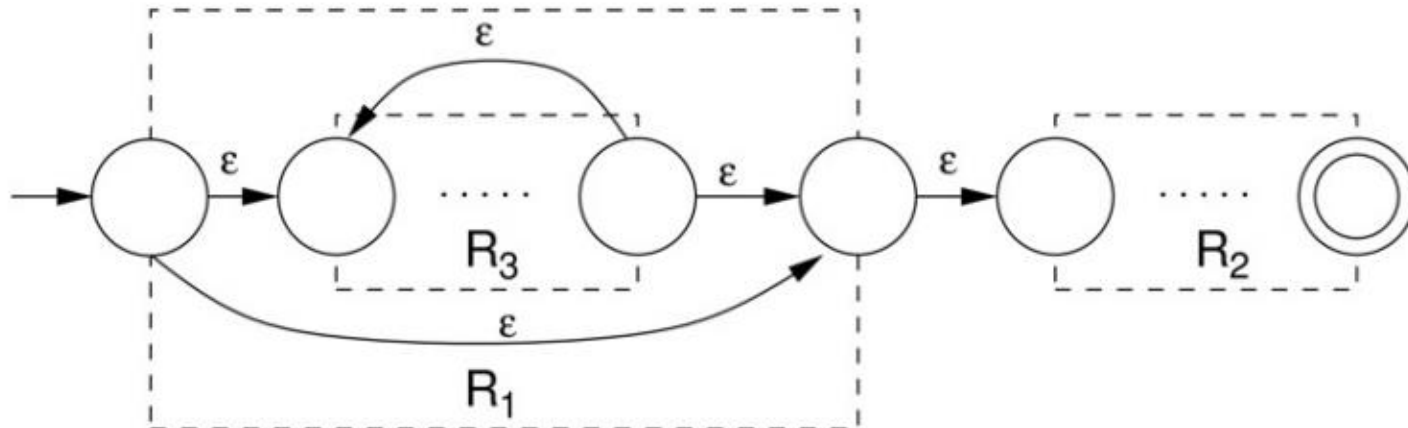
Algoritmo de Thompson

$$R = (0|1)^*0$$

$$R = R_1 R_2$$

$$R_1 = (0|1)^*$$

$$R_1 = R_3^*$$



De expressões regulares para AFND

Algoritmo de Thompson

$$R = (0|1)^*0$$

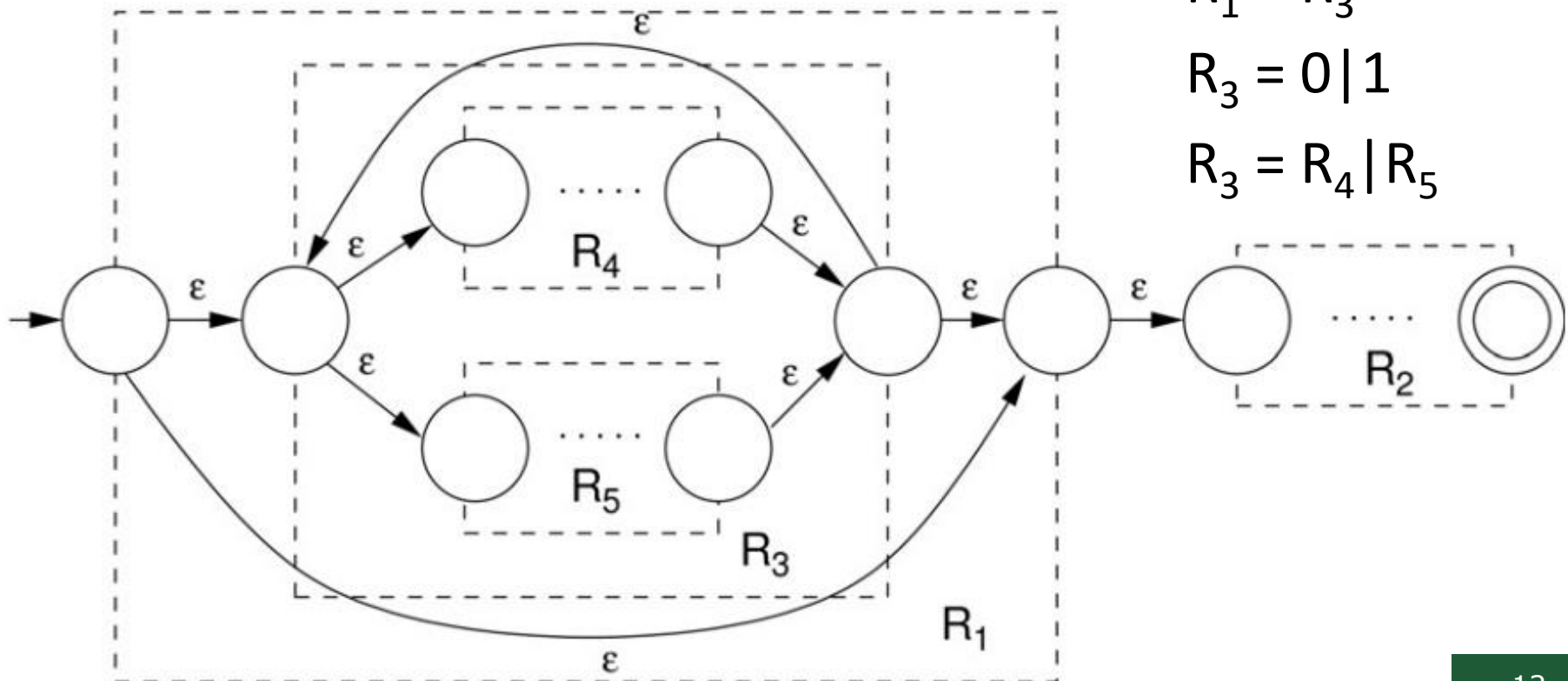
$$R = R_1 R_2$$

$$R_1 = (0|1)^*$$

$$R_1 = R_3^*$$

$$R_3 = 0|1$$

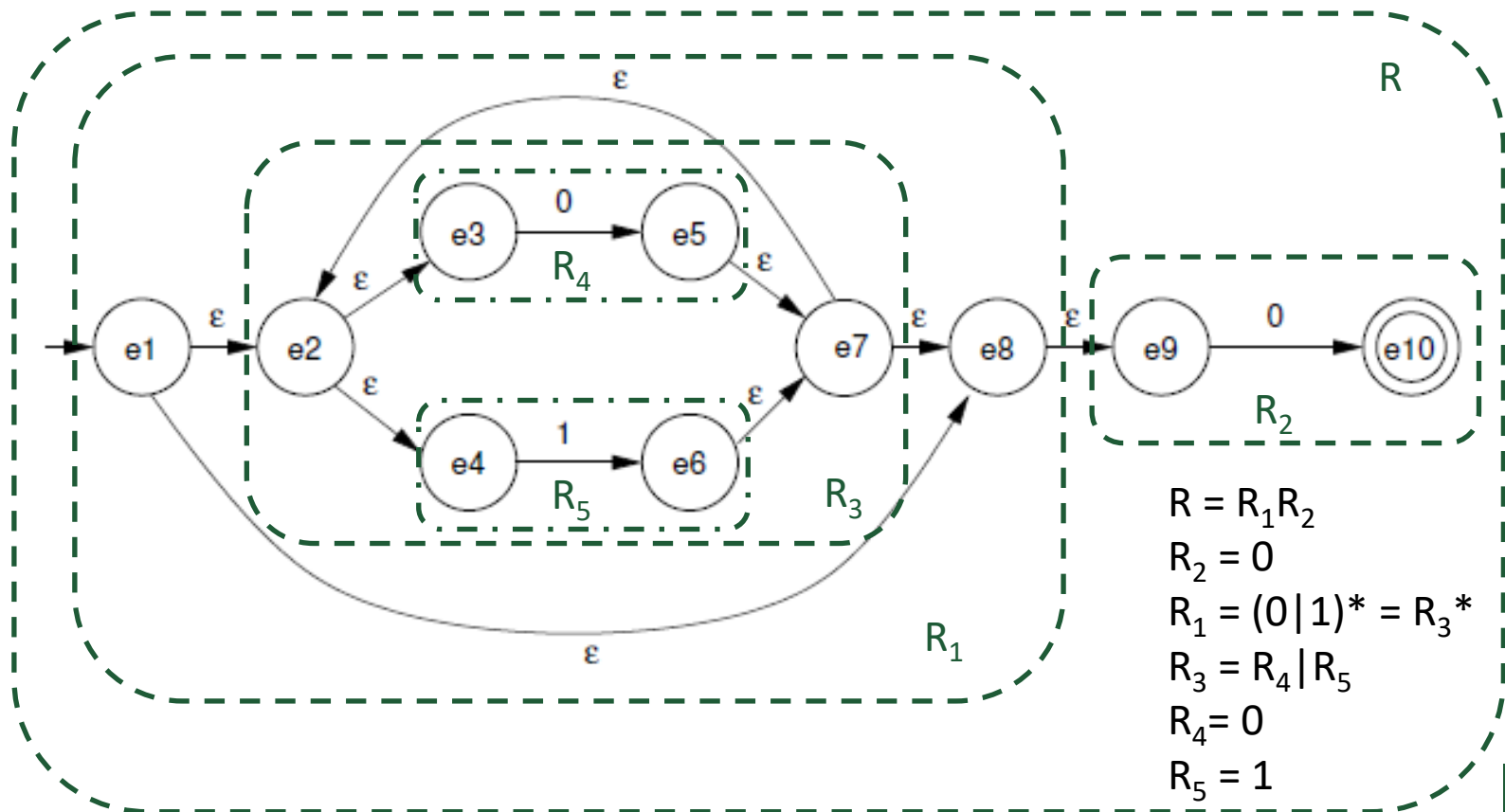
$$R_3 = R_4 | R_5$$



De expressões regulares para AFND

Algoritmo de Thompson

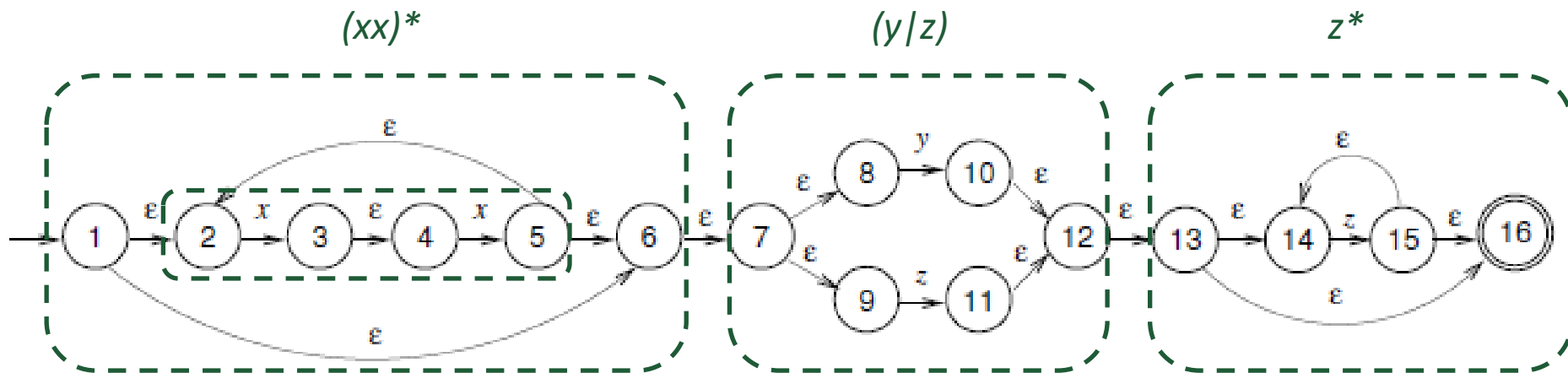
$$R = (0|1)^*0$$



De expressões regulares para AFND

Algoritmo de Thompson

- 1) Dada a expressão regular $(xx)^*(y/z)z^*$ construa o AFND para reconhecer cadeias dessa linguagem.



De expressões regulares para AFND

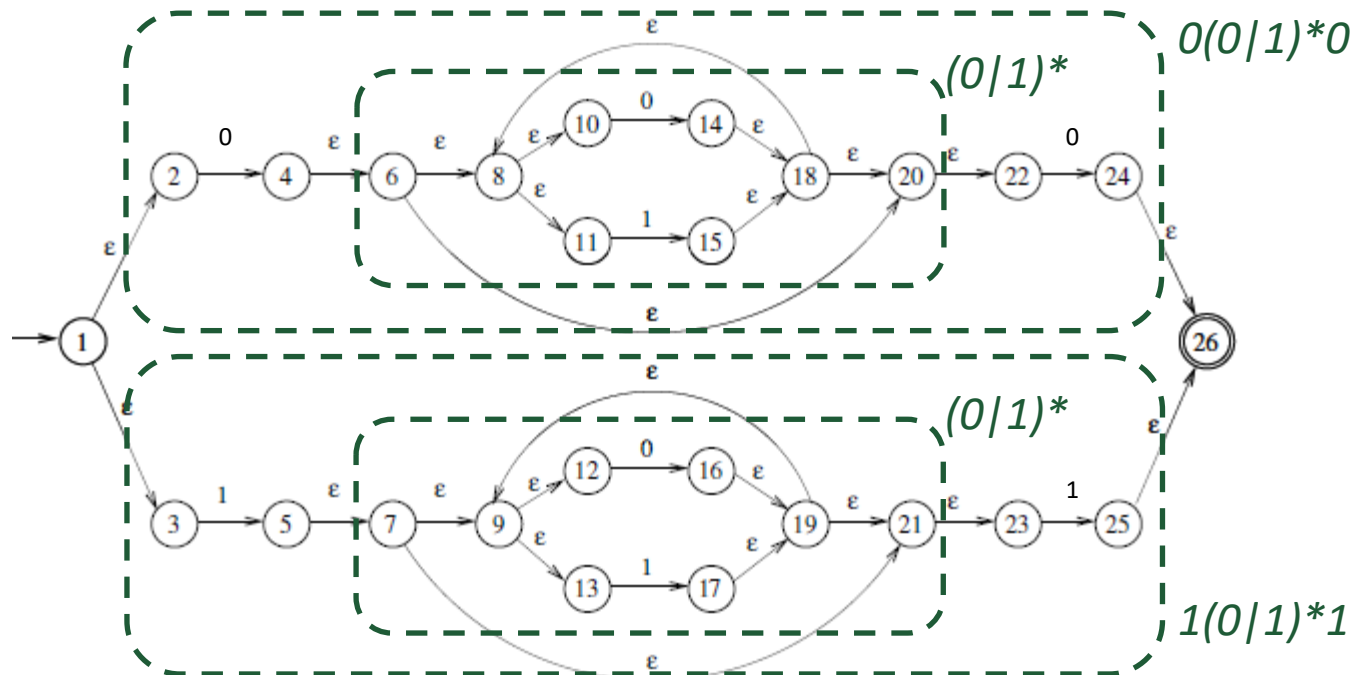
Algoritmo de Thompson

2) Em uma aplicação que aceita cadeias binárias, $\Sigma = \{0, 1\}$, as cadeias aceitas são aquelas que terminam com o mesmo bit que iniciaram.

a) Encontre uma expressão regular que descreva essas cadeias

$0(0/1)^*0/1(0/1)^*1$

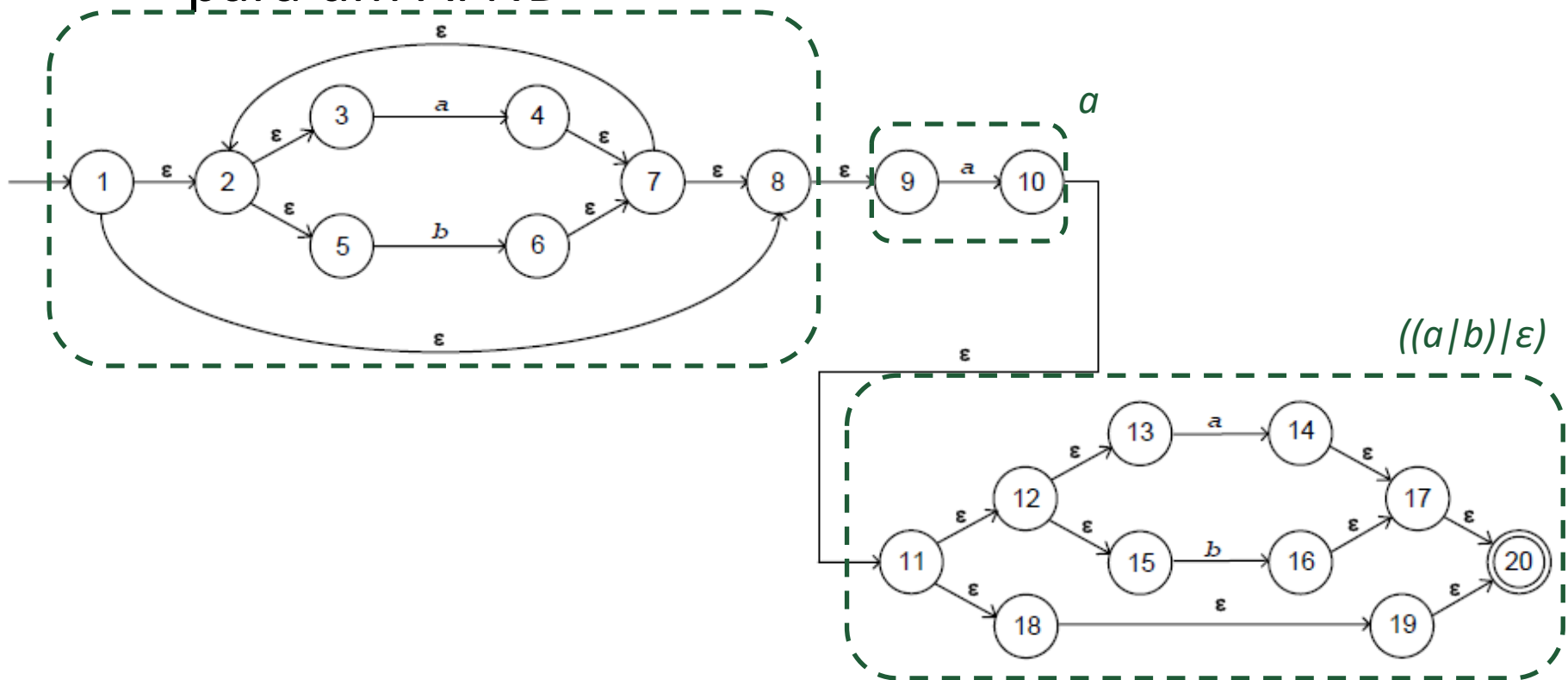
b) Desenvolva o AFND para reconhecer essas cadeias



De expressões regulares para AFND

Algoritmo de Thompson

3) Converter a expressão regular $(a/b)^*a((a/b)/\epsilon)$ para um AFND

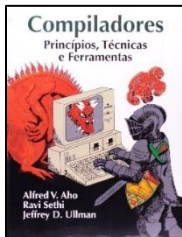


De expressões regulares para AFND

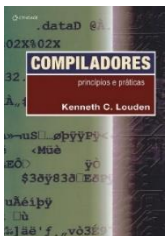
Bibliografia consultada



RICARTE, I. **Introdução à Compilação**. Rio de Janeiro: Editora Campus/Elsevier, 2008.



AHO, A. V.; LAM, M. S.; SETHI, R. e ULLMAN, J. D. **Compiladores: princípios, técnicas e ferramentas**. 2ª edição – São Paulo: Pearson Addison-Wesley, 2008.



LOUDEN, K. C. **Compiladores: princípios e práticas**. São Paulo: Pioneira Thompson Learning, 2004.